J. Symbolic Computation (1992) 14, 283-301

An Optimality Result for Clause Form Translation†

THIERRY BOY DE LA TOUR
Laboratoire d'Informatique Fondamentale
et d'Intelligence Artificielle (IMAG-CNRS)
46, av. Félix Viallet 38031 Grenoble cedex (France)

(*Received* 10 *January* 1989)

The exponential complexity in size of the standard clause form translation is often considered as a serious drawback of the resolution method. Fortunately, a polynomial translation is possible by first introducing definitions, one for each subformula of the conjecture. This exhaustiveness can however be proved inefficient when the length of proofs is considered. In order to improve this interesting technique, we first generalize it to *renamings*, which consist in introducing definitions only for a subset of subformulas, resulting in a wide set of possible clause forms from a single conjecture. We show how a simple and efficient algorithm yields a renaming which, on equivalence-free conjectures, *minimizes* the number of clauses among these clause forms. This translation has been tested on the famous challenge problem by P. Andrews, yielding a spectacular reduction in search space and time, and therefore is one of the more simple and general technique to efficiently produce a resolution proof for this problem.

1. Introduction

The problem of translating a formula into a "good" clause form is known to be a very important one in automated resolution theorem proving. Some theorems have such a huge clause form that it is practically impossible to prove them by resolution. For example, Andrews's challenge problem, which can be found in (Henschen *et al.*, 1980), has only been solved by good clause form translators, rather than by good resolution theorem provers. But it is an exaggeration to incriminate the resolution method as a whole, the problem being only the translation to clause form, with its exponential worst-case complexity, due to the distributivity law.

In order to criticize resolution, one can argue that a refutation does not depend so much on the particular clause form that is obtained from the negation of the conjecture, and that the only way to make resolution efficient is to restrict it. But this argument is not totally correct; on one hand, restrictions are not very useful when the initial set of clauses is big, as illustrated by Andrews's problem. On the other hand, the refutation process may depend drastically on the input clauses: it has been proved in (Tseitin, 1968) that adding clauses, by means of the so-called *extension rule*, can exponentially shorten the length of regular refutations (in the propositional calculus).

† partially supported by ESPRIT2-BRA 3125 Medlar and PRC-IA (MRT-CNRS, France).

0747-7171/92/100283+19 $08.00/0

Hence it is certainly worth spending computation time on the choice of a suitable clause form, if only by repeated applications of the extension rule. The rule is very simple: it consists in adding the clauses corresponding to a *definition of a new propositional symbol*, that is an equivalence between this new symbol and, say, a conjunction of other symbols. However, its actual use is very difficult, since it does not preserve the subformula property: nothing tells how the rule should be applied, or how many times. The extension rule is to the resolution method what the cut rule is to Gentzen systems.

One obvious possibility is to restrict the range of application of the rule, in order to ease the search for relevant definitions. But there is another problem with the extension rule, that it *adds* new clauses, hence the search space necessarily increases. Even though a refutation may be found in fewer steps, the efficiency of the method is doubtful. The problem is that the *use* of the definition is left to the resolution rule, hence delayed, and finally not guaranteed: a refutation may be found before a definition is used. But obviously, a definition can only be relevant if used.

There actually exists a way to enhance the use of definitions simultaneously to their introduction, which is also an elegant way of restricting the range of application of the extension rule; this is *renaming*. The idea is simply to introduce definitions *before* the translation to clause form, only using the formulas at hand to do so — which are obviously the subformulas of the conjecture. This restriction allows one to *replace* in the conjecture those formulas used in the definitions by the corresponding new symbols, which clearly ensures the use of these symbols.

But we still have to search for relevant definitions, which means in this context to search for subformulas to rename. One solution, which was adopted by Tseitin, is simply to rename all the subformulas. This has the advantage of yielding a linear translation to clause form in the propositional calculus. But it has been shown in (Boy de la Tour, 1991) that the renaming of some subformulas necessarily increases the length of the refutation. This is the case for atomic formulas, and negations (provided that the negated formula is renamed).

In (Plaisted and Greenbaum, 1986) there is a translation to clause form, called *structure preserving*, where all the subformulas are renamed except negations and atomic formulas. Despite its very interesting worst-case complexity, this translation has some drawbacks, see (Boy de la Tour and Chaminade, 1990), one of which is that it still performs some renamings that have been proved useless in (Boy de la Tour, 1991).

The solution is therefore not that simple, and the aforementioned results on useless renamings do not lead to any regular criterion for useful renamings. However, all these useless renamings (including the renaming of negations and atomic subformulas) have one thing in common: they also result in an increase of the number of clauses. Although the correspondence between the subformulas that increase the length of the refutation and those that increase the number of clauses when renamed may not be exact, it seems interesting to focus on this last criterion, since it is easy to compute, as shown in the following.

The next section presents the general method we use to decrease the number of clauses: it simply consists in renaming a subformula iff the resulting number of clauses decreases. Applying this criterion in sequence to the subformulas of the conjecture, in any order, and until there is no subformula left that satisfies the test, is already an efficient method and leads to a result of restricted optimality: the number of clauses is less than both the standard and the structure preserving clause forms. The complexity in size of the method, independent of the search, is quickly deduced.

In section 3, we consider a particular case of this method, corresponding to top-down orderings of the subformulas, which of course satisfy all the properties shown in section 2. We then prove that this method, when equivalence-free conjectures are considered, is *optimal* in number of clauses, comparing with all possible renamings of the conjecture. An example shows that optimality is not guaranteed in the general, non equivalence-free case.

Last, we have applied this top-down transformation on examples which are known to be difficult to translate into concise clause form and efficiently obtained very good results in this respect. This is presented in section 4.

2. Definitions and First Results

2.1. DEFINITIONS

The syntax of the formulas includes, in addition to standard features, undetermined arity for conjunction and disjunction — this allows one to consider conjunctive normal forms as formulas, instead of lists of lists of literals — quantification over lists of variables, and predicate symbols with arity 0, thus containing the syntax of propositional formulas.

$SF(\psi)$ denotes the *multiset* of subformulas of ψ. This formalization allows to speak about the tree structure of a formula, without the burden of tree occurrences. Burden which would come from the fact that renaming transformations do change occurrences of subformulas. No notation of these changes is needed since they are always the same.

The number of subformulas of ψ, $|SF(\psi)|$, is therefore the number of nodes (including root and leaves) of ψ considered as a labelled tree.

$SF^*(\psi)$ stands for $SF(\psi) - \{\psi\}$, $\varphi \sqsubseteq \psi$ for $\varphi \in SF(\psi)$ (we say that ψ is a *supformula* of φ), and $\varphi \sqsubset \psi$ for $\varphi \in SF^*(\psi)$. We call *direct subformulas* of ψ the maximal elements of $SF^*(\psi)$ (in the order $\sqsubseteq$). $FV(\psi)$ is the set of free variables of ψ.

We will use a slight modification of the well-known notion of *polarity* of a subformula. If $\varphi \sqsubseteq \psi$, in the case there is no equivalence in the interval $]\varphi, \psi]$, the definition of $pol(\varphi, \psi)$ is standard: its value is 1 (resp. -1) if the number of negations in $]\varphi, \psi]$ plus the number of implications $\varphi_1 \Rightarrow \varphi_2$ in $]\varphi, \psi]$ such that $\varphi \sqsubseteq \varphi_1$ is even (resp. odd). Otherwise, we define it to be 0. Polarities will be designated by the meta-variable s.

We first consider the transformation into clause form which is to be applied after the renaming. This transformation involves linearization (elimination of equivalences), skolemisation, prenex and conjunctive normalizations: we will note it *clausal*.

The last three transformations are standard, the first involves a particularity which has revealed itself indispensable to obtain our results: following (Henschen *et al.*, 1980), the equivalences $p \Leftrightarrow q$ are considered in a top-down order, and transformed depending on their polarity in the formula. If $p \Leftrightarrow q$ has positive polarity, it is transformed into $(p \Rightarrow q) \wedge (q \Rightarrow p)$, otherwise into $(p \wedge q) \vee (\neg p \wedge \neg q)$. More precisely, this linearization is $lin(\varphi, +1)$, where $lin(\varphi, s)$ is inductively defined: if φ is atomic, $lin(\varphi, s) = \varphi$; if $\varphi = \varphi_1 \Leftrightarrow \varphi_2$, then

$$lin(\varphi, s) = \begin{cases} (lin(\varphi_1, -1) \Rightarrow lin(\varphi_2, +1)) \wedge (lin(\varphi_2, -1) \Rightarrow lin(\varphi_1, +1)) & \text{if } s = +1 \\ (lin(\varphi_1, -1) \wedge lin(\varphi_2, -1)) \vee (\neg lin(\varphi_1, +1) \wedge \neg lin(\varphi_2, +1)) & \text{if } s = -1 \end{cases}$$

otherwise, $lin(\varphi, s)$ is the formula φ where each direct subformula φ' of φ is replaced by $lin(\varphi', s.pol(\varphi', \varphi))$.

This performs simplifications which are not recognizable if the polarity is not consid-

Table 1. Number of clauses

ψ	$p(\psi)$	$\overline{p}(\psi)$
$\psi_1 \wedge \cdots \wedge \psi_n$	$\sum_{i=1}^n p(\psi_i)$	$\prod_{i=1}^n \overline{p}(\psi_i)$
$\psi_1 \vee \cdots \vee \psi_n$	$\prod_{i=1}^n p(\psi_i)$	$\sum_{i=1}^n \overline{p}(\psi_i)$
$\psi_1 \Rightarrow \psi_2$	$\overline{p}(\psi_1)p(\psi_2)$	$p(\psi_1) + \overline{p}(\psi_2)$
$\psi_1 \Leftrightarrow \psi_2$	$p(\psi_1)\overline{p}(\psi_2) + \overline{p}(\psi_1)p(\psi_2)$	$p(\psi_1)p(\psi_2) + \overline{p}(\psi_1)\overline{p}(\psi_2)$
$Qx_1..x_n.\psi_1$	$p(\psi_1)$	$\overline{p}(\psi_1)$
$\neg\psi_1$	$\overline{p}(\psi_1)$	$p(\psi_1)$
atomic	1	1

Table 2. Coefficients

φ	$a^{\psi}_{\varphi_i}$	$b^{\psi}_{\varphi_i}$
$\varphi_1 \wedge \cdots \wedge \varphi_n$	a^{ψ}_{φ}	$b^{\psi}_{\varphi} \prod_{j \neq i} \overline{p}(\varphi_j)$
$\varphi_1 \vee \cdots \vee \varphi_n$	$a^{\psi}_{\varphi} \prod_{j \neq i} p(\varphi_j)$	b^{ψ}_{φ}
$\varphi_1 \Rightarrow \varphi_2, i = 1$	b^{ψ}_{φ}	$a^{\psi}_{\varphi} p(\varphi_2)$
$\varphi_1 \Rightarrow \varphi_2, i = 2$	$a^{\psi}_{\varphi}\overline{p}(\varphi_1)$	b^{ψ}_{φ}
$\varphi_1 \Leftrightarrow \varphi_2, j = 3 - i$	$a^{\psi}_{\varphi}\overline{p}(\varphi_j) + b^{\psi}_{\varphi}p(\varphi_j)$	$a^{\psi}_{\varphi}p(\varphi_j) + b^{\psi}_{\varphi}\overline{p}(\varphi_j)$
$\neg\varphi_i$	b^{ψ}_{φ}	a^{ψ}_{φ}
$Qx_1..x_n.\varphi_i$	a^{ψ}_{φ}	b^{ψ}_{φ}
$\varphi_i = \psi$	1	0

ered, and the formula transformed into clause form, hence this technique decreases the number of clauses. It is very useful, both practically and theoretically (the rest of the paper would have been much more complicated without it).

Considering these transformations, it is easy to compute the number of clauses obtained from a formula ψ, which we note $p(\psi)$ (see table 1, $\overline{p}(\psi)$ stands for $p(\neg\psi)$), without computing *clausal*(ψ). This would not have been the case considering simplifications and/or subsumptions, since then the tree structure of the clause form would be dependent on non-logical symbols. With *clausal* as it is, we have in particular the important property that the number of clauses of a formula depends only on the number of clauses of its subformulas (and/or their negations), out of which we state the following definition: $\forall\varphi \sqsubseteq \psi$, let P^{ψ}_{φ} and $\overline{P}^{\psi}_{\varphi}$ be the functions from $\mathbb{N}^* \times \mathbb{N}^*$ to $\mathbb{N}^*$ defined by:

$$\forall\varphi', P^{\psi}_{\varphi}(p(\varphi'), \overline{p}(\varphi')) = p(\psi[\varphi \leftarrow \varphi']) \text{ and } \overline{P}^{\psi}_{\varphi}(p(\varphi'), \overline{p}(\varphi')) = \overline{p}(\psi[\varphi \leftarrow \varphi'])$$

EXAMPLE 2.1. Let $\psi = (\varphi \Rightarrow \varphi_1) \wedge \varphi_2$, where φ_1 and φ_2 are any formulas, and φ is the "variable" formula; we are considering the variation of $p(\psi)$ with respect to the variation of φ. We have $p(\psi) = \overline{p}(\varphi)p(\varphi_1) + p(\varphi_2)$ and $\overline{p}(\psi) = (p(\varphi) + \overline{p}(\varphi_1))\overline{p}(\varphi_2)$, hence $P^{\psi}_{\varphi}(x, y) = p(\varphi_1)y + p(\varphi_2)$ and $\overline{P}^{\psi}_{\varphi}(x, y) = \overline{p}(\varphi_2)x + \overline{p}(\varphi_2)\overline{p}(\varphi_1)$.

From table 1, it is easy to see that $P^{\psi}_{\varphi}(x, y)$ is always of the form $a^{\psi}_{\varphi}x + b^{\psi}_{\varphi}y + c^{\psi}_{\varphi}$ with $a^{\psi}_{\varphi} \geq 0$ ($a^{\psi}_{\varphi} = 0$ iff $pol(\varphi, \psi) = -1$), $b^{\psi}_{\varphi} \geq 0$ ($b^{\psi}_{\varphi} = 0$ iff $pol(\varphi, \psi) = 1$) and $c^{\psi}_{\varphi} \geq 0$. If φ is a direct subformula of $\varphi' \sqsubseteq \psi$, then a^{ψ}_{φ} and b^{ψ}_{φ} can be computed from $a^{\psi}_{\varphi'}$ and $b^{\psi}_{\varphi'}$, and also from the values $p(\varphi_i), \overline{p}(\varphi_i)$, where the φ_i's are the other direct subformulas of φ'; see table 2.

To define the renaming transformation, we first associate to each subformula $\varphi \sqsubseteq \psi$ a new predicate symbol, noted SkP^{ψ}_{φ} (Skolem predicate), and a new literal $SkP^{\psi}_{\varphi}(x_1..x_n)$, noted SkL^{ψ}_{φ}, where $x_1..x_n$ are the free variables of φ. We call *definition of* SkL^{ψ}_{φ}, and note $Def_{\psi}(\varphi)$ the following formula, depending on the polarity of φ in ψ: if $pol(\varphi,\psi) = 1$, then $Def_{\psi}(\varphi) = \forall x_1..x_n.SkL^{\psi}_{\varphi} \Rightarrow \varphi$; if $pol(\varphi,\psi) = -1$, then $Def_{\psi}(\varphi) = \forall x_1..x_n.\varphi \Rightarrow SkL^{\psi}_{\varphi}$; and $Def_{\psi}(\varphi) = \forall x_1..x_n.SkL^{\psi}_{\varphi} \Leftrightarrow \varphi$ if $pol(\varphi,\psi) = 0$. The rule is that $pol(\varphi, Def_{\psi}(\varphi)) = pol(\varphi,\psi)$.

A *renaming* R of ψ is simply a submulti-set of $SF^*(\psi)$. For technical reasons, we do not allow the renaming of ψ, but this would clearly be useless. To each $\varphi \sqsubseteq \psi$ we associate a particular submulti-set of R, noted $Inf_R(\varphi)$, which is $\max\{\varphi' \in R \,/\, \varphi' \sqsubset \varphi\}$. $Inf_R(\varphi)$ is a renaming of φ. We also associate to φ a particular element of R, if $\{\varphi' \in R \,/\, \varphi \sqsubseteq \varphi'\} \neq \emptyset$, noted $Sup_R(\varphi)$, which is $\min\{\varphi' \in R \,/\, \varphi \sqsubseteq \varphi'\}$.

$\varphi[R]$ denotes the formula $\varphi[\varphi' \leftarrow SkL^{\psi}_{\varphi'}]_{\varphi' \in Inf_R(\varphi)}$. To each renaming R of ψ we associate the formula noted $Rnm(R,\psi)$, which is $\psi[R] \wedge \bigwedge_{\varphi \in R} Def_{\psi}(\varphi[R])$. It is proved in (Boy de la Tour, 1991) that the preservation of satisfiability under any renaming holds (as well as under any skolemisation: this is a trivial consequence of a general theorem on conservative expansions).

2.2. A FUNDAMENTAL RESULT

Given a formula ψ, we now have one clause form for each renaming R of ψ, namely $clausal(Rnm(R,\psi))$. Let $struct\_pres(\psi)$ be $SF^*(\psi)$ except for negations and atomic subformulas. It is shown in (Plaisted and Greenbaum, 1986) that the length of the *structure preserving* clause form $clausal(Rnm(struct\_pres(\psi),\psi))$ is bounded by $O((1+V(\psi))|\psi|)$, where $V(\psi) = \max\{|FV(\varphi)| / \varphi \sqsubseteq \psi\}$, and $|\psi|$ is the length of ψ. It can also be proved that the number of clauses is linear in $|SF(\psi)|$, the number of subformulas of ψ. This is obviously not the case for any renaming; in particular, for $R = \emptyset$, which yields the standard clause form, the number of clauses (and the length itself) is exponential in $|SF(\psi)|$. However, it is proved in (Boy de la Tour, 1991) that the length of the clauses is bounded by $O((1+V(\psi))|\psi|)$, which shows the interest of focusing only on the number of clauses — if the exponential complexity is to be avoided, which is certainly the case.

For this purpose, we define the *benefit* of a renaming R in a formula ψ, noted $B(R,\psi)$, to be $p(\psi) - p(Rnm(R,\psi))$. $B(\varphi,\psi)$ stands for $B(\{\varphi\},\psi)$. If $\varphi[R] \sqsubset \psi[R]$, $B(\varphi,\psi[R])$ stands for $B(\varphi[R],\psi[R])$, and for $B(SkL^{\psi}_{\varphi},\psi[R])$ if $\varphi \in Inf_R(\psi)$. We notice that if $b^{\psi}_{\varphi} = 0$ (i.e. $pol(\varphi,\psi) = 1$), then $B(\varphi,\psi) = (a^{\psi}_{\varphi} - 1)(p(\varphi) - 1) - 1$, hence that $B(\varphi,\psi) \geq 0$ iff $a^{\psi}_{\varphi} > 1 \wedge p(\varphi) > 1$. This will be of constant use.

We are obviously interested in those renamings R such that $B(R,\psi) \geq 0$, but we can restrict ourselves to a smaller subset of renamings using the following fundamental theorem of monotony (in short *ftm*).

Theorem 2.1. *$\forall\psi, \forall\varphi \sqsubset \psi$, if R, R' are renamings of ψ such that $R \subset R'$, then*

$$B(\varphi, Rnm(R,\psi)) \geq B(\varphi, Rnm(R',\psi))$$

Proof. We show that $\forall\varphi' \sqsubset \psi, B(\varphi,\psi) \geq B(\varphi, Rnm(\varphi',\psi))$, from which the general case can be deduced. We have $B(\varphi,\psi) = P_\varphi^\psi(p(\varphi),\overline{p}(\varphi)) - P_\varphi^\psi(1,1) - p(Def_\psi(\varphi))$, and we distinguish three cases:

1 if φ and φ' are disjoint, then

$$\begin{aligned} B(\varphi, Rnm(\varphi',\psi)) &= B(\varphi,\psi[\varphi']) \\ &= P_\varphi^{\psi[\varphi']}(p(\varphi),\overline{p}(\varphi)) - P_\varphi^{\psi[\varphi']}(1,1) - p(Def_\psi(\varphi)) \end{aligned}$$

The coefficients of the polynomial $P_\varphi^{\psi[\varphi']}$ are smaller than those of P_φ^ψ (because $p(\psi[\varphi']) \le p(\psi)$), hence $B(\varphi,\psi[\varphi']) \le B(\varphi,\psi)$ (because $p(\varphi),\overline{p}(\varphi),p(\varphi'),\overline{p}(\varphi') \ge 1$, and the coefficients of $P_\varphi^{\psi[\varphi']}$ are positive).

2 if $\varphi \sqsubseteq \varphi'$ then

$$\begin{aligned} B(\varphi, Rnm(\varphi',\psi)) &= B(\varphi, Def_\psi(\varphi')) \\ &= P_\varphi^{Def_\psi(\varphi')}(p(\varphi),\overline{p}(\varphi)) - P_\varphi^{Def_\psi(\varphi')}(1,1) - p(Def_\psi(\varphi)) \end{aligned}$$

The coefficients of $P_\varphi^{Def_\psi(\varphi')}$ are smaller than those of P_φ^ψ, because $P_\varphi^\psi(x,y) = P_{\varphi'}^\psi(P_\varphi^{\varphi'}(x,y),\overline{P}_\varphi^{\varphi'}(x,y))$ and $P_\varphi^{Def_\psi(\varphi')}(x,y) = P_{\varphi'}^{Def_\psi(\varphi')}(P_\varphi^{\varphi'}(x,y),\overline{P}_\varphi^{\varphi'}(x,y))$, and the coefficients of $P_{\varphi'}^{Def_\psi(\varphi')}$ are obviously smaller than those of $P_{\varphi'}^\psi$. Hence, as above, $B(\varphi, Def_\psi(\varphi')) \le B(\varphi,\psi)$.

3 if $\varphi' \sqsubseteq \varphi$, since we always have

$$B(\varphi,\psi) + B(\varphi', Rnm(\varphi,\psi)) = B(\{\varphi,\varphi'\},\psi) = B(\varphi',\psi) + B(\varphi, Rnm(\varphi',\psi))$$

then $B(\varphi, Rnm(\varphi',\psi)) \le B(\varphi,\psi)$ is equivalent to $B(\varphi', Rnm(\varphi,\psi)) \le B(\varphi',\psi)$, which is true according to the previous case (reversing φ and φ').

□

2.3. RESTRICTED OPTIMALITY

It is then clear that the benefit of a subformula depends on previous renamings, hence that renamings of positive benefit have to be computed in *sequence*; considering a renaming R such that $B(R,\psi) \ge 0$, if there is a sequence $(\varphi_1..\varphi_n)$ such that $R = \{\varphi_1..\varphi_n\}$ and $\exists i, B(\varphi_i, Rnm(\{\varphi_1..\varphi_{i-1}\},\psi)) < 0$, then we have from *ftm* that $B(\varphi_i, Rnm(R - \{\varphi_i\},\psi)) < 0$. Hence $R - \{\varphi_i\}$ is more interesting than R, and we can restrict ourselves to *positive* renamings, corresponding to sequences $(\varphi_1..\varphi_n)$ such that $\forall i, B(\varphi_i, Rnm(\{\varphi_1..\varphi_{i-1}\},\psi)) \ge 0$.

Obviously, we must not leave any subformula of positive benefit unrenamed, hence we also aim at *complete* renamings R, verifying $\forall\varphi \sqsubset \psi, B(\varphi, Rnm(R,\psi)) < 0$. It is quite easy to compute positive and complete renamings:

$$\begin{aligned} R_{inf}(\psi) = \ &\textbf{begin} \\ &\quad R := \emptyset\ ; \\ &\quad \textbf{while}\ \exists\varphi \sqsubset \psi, B(\varphi, Rnm(R,\psi)) \ge 0\ \textbf{do}\ R := R \cup \{\varphi\}\ ; \\ &\quad \textbf{return}(R) \\ &\textbf{end} \end{aligned}$$

This procedure is non-deterministic, since the strategy for choosing the subformulas

$\varphi \sqsubset \psi$ in the loop is not specified. Depending on this search, we can obtain any positive and complete renaming, and we are sure that all *optimal* renamings (those that minimize the function $\lambda R.p(Rnm(R, \psi))$) are positive and complete. The converse is however false (see examples 3.1 and 3.2).

The problem is then to find a good search to reach the best possible renaming with respect to the number of clauses. Another possibility is to compute them all, but this would certainly lead to an exponential and practically inefficient procedure. We will restrict ourselves to the procedure R_{inf} (no backtracking).

Before considering any particular search, we can first state some general properties about positive and complete renamings. We actually have a property of restricted optimality, relative to the standard and the structure preserving clause forms.

THEOREM 2.2.

$$\forall \psi, p(Rnm(R_{inf}(\psi), \psi)) \leq min(p(\psi), p(Rnm(struct\_pres(\psi), \psi)))$$

PROOF. The first inequality $p(Rnm(R_{inf}(\psi), \psi)) \leq p(\psi)$ is trivial. To prove the second one, $p(Rnm(R_{inf}(\psi), \psi)) \leq p(Rnm(struct\_pres(\psi), \psi))$, it is sufficient according to *ftm*, since $R_{inf}(\psi)$ is complete, to build a renaming $R \supset R_{inf}(\psi)$ such that $p(Rnm(R, \psi)) = p(Rnm(struct\_pres(\psi), \psi))$.

We construct R by adding to $R_{inf}(\psi)$ all the other subformulas of ψ, excepting of course negations and atomic formulas. As the test $B(\varphi, \psi) \geq 0$ is not satisfied on atomic formulas, $R_{inf}(\psi)$ does not contain any. However, $R_{inf}(\psi)$ may contain negations $\neg\varphi$, but it is easy to see that $B(\varphi, Rnm(\neg\varphi, \psi)) < 0$, hence according to *ftm* we then have $\varphi \notin R_{inf}(\psi)$, which allows in that case not to include φ in R either. More precisely, we have $\forall \varphi \sqsubset \psi, \varphi \in struct\_pres(\psi) - R \Leftrightarrow \neg\varphi \in R - struct\_pres(\psi)$. Therefore, we have $R \supset R_{inf}(\psi)$, differing from $struct\_pres(\psi)$ only for some negations and their arguments. But it is then easy to see that renaming either a negation or its argument leads to the same number of clauses, hence $p(Rnm(R, \psi)) = p(Rnm(struct\_pres(\psi), \psi))$. □

2.4. COMPLEXITY

As an important consequence, the number of clauses obtained from any $R_{inf}(\psi)$ is bounded by a linear function of $|SF(\psi)|$. From this we can conclude, by mixing some complex and boring upper bound lemmas, see (Boy de la Tour, 1991), that the length of these clause forms is bounded by $O((1 + V(\psi))|\psi||SF(\psi)|)$, hence a greater complexity than the structure preserving clause form, which is bounded by $O((1 + V(\psi))|\psi|)$. The difference is easily explained with the following

EXAMPLE 2.2. Consider the formula $F_n = (P_1 \wedge \cdots \wedge P_n) \vee Q_1 \vee \cdots \vee Q_n$, we have $p(F_n) = n$, and no subformula $\varphi \sqsubset F_n$ satisfies the test $B(\varphi, F_n) \geq 0$; hence n is the optimal number of clauses. But its clause form is

$$cnf(F_n) = (P_1 \vee Q_1 \vee \cdots \vee Q_n) \wedge (P_2 \vee Q_1 \vee \cdots \vee Q_n) \wedge \cdots \wedge (P_n \vee Q_1 \vee \cdots \vee Q_n)$$

the length of which is in $O(n^2)$. In comparison, we have $struct\_pres(F_n) = \{\varphi_n\}$, where $\varphi_n = P_1 \wedge \cdots \wedge P_n$, and the structure preserving clause form is

$$(SkP^{\psi}_{\varphi_n} \vee Q_1 \vee \cdots \vee Q_n) \wedge (\neg SkP^{\psi}_{\varphi_n} \vee P_1) \wedge \cdots \wedge (\neg SkP^{\psi}_{\varphi_n} \vee P_n)$$

the length of which is linear in n, but has $n + 1$ clauses.

This proves that, *independently of the search for subformulas*, the worst-case complexity in size of $clausal(Rnm(R_{inf}(\psi),\psi))$ is exactly $O((1+V(\psi))|\psi||SF(\psi)|)$ — it is very easy to add variables and existential quantifiers to this example in order to obtain, by skolemisation, the factor $1+V(\psi)$. However, the average complexity of these translations seems better than the structure preserving clause form: on real examples, we did not find a greater length with R_{inf} than with *struct_pres*.

3. Linear Optimality

3.1. TOP-DOWN RENAMINGS

We can now take care of the order in which subformulas are going to be searched. We may adopt a bottom-up search, which seems more natural. But it appears that a top-down search is very likely to bring fewer clauses, as suggested by the following

EXAMPLE 3.1. Consider the formula $\psi = \varphi\vee\varphi_1$ with $\varphi = A\wedge(B\vee\varphi_2)$, and φ_1, φ_2 such that $p(\varphi_1) \geq 2$, $p(\varphi_2) \geq 2$. We have $a^{\psi}_{\varphi_2} = p(\varphi_1)$, hence $B(\varphi_2, \psi) \geq 0$. But $p(\varphi[\varphi_2]) = 2$, hence $B(\varphi, \psi[\varphi_2]) \geq 0$, and therefore a bottom-up search renames φ and φ_2. However, we have $B(\varphi,\psi) \geq 0$, and $B(\varphi_2, Def_\psi(\varphi)) < 0$, hence a top-down search does not rename φ_2, and yields fewer clauses.

The purpose of this section is to strengthen the previous result of restricted optimality by imposing a particular search within the tree structure of the conjecture. We will actually prove the full optimality — among all possible renamings — of renamings obtained from R_{inf} with top-down strategies, with the restriction that the conjecture should be linear (that is equivalence-free).

The meaning of top-down strategies for the procedure R_{inf} should be clear, but in the following, we will rather work on top-down *renamings*. This is quite different since we consider renamings as *multi-sets* of subformulas, instead of sequences, as would be more convenient to formalize a search. The definition of these renamings will seem more restrictive, and we will show in section 3.4 the correspondence between the two.

We call *renaming of φ free in ψ* any renaming R of $\varphi \sqsubseteq \psi$ such that $\forall\xi \in R$, $B(\xi, Rnm(R-\{\xi\},\psi)) \geq 0$. This means that the formulas $\xi \in R$ can be obtained from the procedure R_{inf} *in any order*, since we then have, as a consequence of *ftm*, $\forall R' \subset R-\{\xi\}, B(\xi, Rnm(R',\psi)) \geq 0$.

Now, we call *renaming of φ top-down in ψ* any renaming R of φ free in ψ and such that $\forall\xi \sqsubseteq \varphi$, if $Sup_R(\xi)$ exists, then $B(\xi, Def_\psi(Sup_R(\xi))[R-SF(\xi)]) < 0$, and else $B(\xi, \psi[R-SF(\xi)]) < 0$. If $\varphi = \psi$, we simply say that R is top-down in ψ.

The last condition indicates the top-down saturation: no subformula is renamed — is in R — if it has a supformula of positive benefit which is not renamed. We see that this *saturation condition* is written in a simplified way, supposing that all subformulas in $R-SF(\xi)$ have already been replaced by their Skolem literals. This is not always the case considering the order in which R_{inf} performs these replacements, and this is the reason why it is much simpler to consider multi-sets instead of sequences of subformulas, hence the necessity to add the condition that R is free, and the order irrelevant. This restriction is not a limitation as long as we are interested in optimal renamings, since they are free: let R be a non free renaming, then $\exists\varphi \in R, B(\varphi, Rnm(R-\{\varphi\},\psi)) < 0$, hence $p(Rnm(R-\{\varphi\},\psi)) < p(Rnm(R,\psi))$, and R is not optimal.

It is rather easy to show that any top-down renaming is positive and complete: every renaming free in ψ is positive, and the condition of top-down saturation implies that no subformula in $SF(\psi) - R$ has positive benefit in $Rnm(R,\psi)$. However, in the general case, top-down strategies are not necessarily optimal, as shown by the following

EXAMPLE 3.2. Let $\psi = \varphi \Leftrightarrow (A \wedge B)$, with $\varphi = \varphi' \vee (A' \wedge B')$. We have $B(\varphi,\psi) = \overline{p}(\varphi') - 2$ and $B(\varphi', Def_\psi(\varphi)) = p(\varphi') - 3$, hence if we chose φ' such that $p(\varphi'), \overline{p}(\varphi')$ are great enough, we have a top-down renaming of ψ containing φ and φ'. However, we have $B(\varphi[\varphi'], \psi[\varphi']) = -1$, hence $B(\varphi', \psi) > B(\{\varphi, \varphi'\}, \psi)$, and this top-down renaming, containing φ, is not free in ψ, hence not optimal.

3.2. INVARIANCE

From here and to the end of section 3, we will restrict ourselves to linear formulas. For the sake of simplicity, we will work on negation normal forms (nnf), which is not a problem since transformation into nnf (from a linear formula) is linear in time. We then have polynomials P_φ^ψ with only one argument. However, the following results are valid on all linear formulas, not only nnfs.

We will first show that all top-down renamings result in the same number of clauses, the consequence of which, from a computational point of view, is that we do not have to program a particular search: any top-down search is appropriate, and in particular the simple depth-first, left-right. Surprisingly so, the proof of this fact is more difficult and technical than (the rest of) the proof of optimality. We first have to establish, as a lemma, a structural property of top-down renamings.

LEMMA 3.1. *Let $\varphi \sqsubseteq \psi$, R a renaming of φ top-down in ψ, $\varphi' \sqsubseteq \varphi$, $R' = R \cap SF^*(\varphi')$, and if $Sup_R(\varphi')$ exists then $\psi' = Def_\psi(Sup_R(\varphi')[R - R'])$ else $\psi' = \psi[R - R']$, then R' is a renaming of φ' top-down in ψ'.*

PROOF. We first show that R' is free in ψ': $\forall \xi \in R'$, let $e = B(\xi, Rnm(R' - \{\xi\}, \psi'))$. If $\xi \notin Inf_{R'}(\varphi')$, the benefit of ξ does not change: $\exists \xi' \in R'$ such that $\xi \in Inf_{R'}(\xi')$ and we have $e = B(\xi, Def_\psi(\xi'[R' - \{\xi\}])) = B(\xi, Def_\psi(\xi'[R - \{\xi\}]))$ since $Inf_R(\xi') = Inf_{R'}(\xi')$, hence $e = B(\xi, Rnm(R - \{\xi\}, \psi)) \geq 0$ since R is free in ψ. If $\xi \in Inf_{R'}(\varphi')$, then $e = B(\xi, \psi'[R' - \{\xi\}]) = B(\xi, Def_\psi(Sup_R(\varphi')[R - R'][R' - \{\xi\}]))$ (if $Sup_R(\varphi')$ exists). But $\xi \in R'$, thus $(R - R') \cup (R' - \{\xi\}) = R - \{\xi\}$, hence $e = B(\xi, Def_\psi(Sup_R(\varphi')[R - \{\xi\}]))$, but $Sup_R(\varphi') = Sup_R(\xi)$, hence $e = B(\xi, Rnm(R - \{\xi\}, \psi)) \geq 0$. The case where $Sup_R(\varphi')$ does not exist is similar.

We now have to show that the saturation condition holds: $\forall \xi \sqsubseteq \varphi'$, if $Sup_{R'}(\xi)$ exists, we have $Sup_{R'}(\xi) = Sup_R(\xi) \sqsubseteq \varphi'$, thus $Sup_{R'}(\xi)[R'] = Sup_R(\xi)[R]$, hence $B(\xi, Def_\psi(Sup_{R'}(\xi)[R' - SF(\xi)])) = B(\xi, Def_\psi(Sup_R(\xi)[R - SF(\xi)])) < 0$ since R is top-down in ψ. Otherwise, let $e = B(\xi, \psi'[R' - SF(\xi)])$, in the case where $Sup_R(\varphi')$ does not exists, then $e = B(\xi, \psi[R - R'][R' - SF(\xi)])$. But $SF(\xi) \cap (R - R') = \emptyset$, hence $e = B(\xi, \psi[R - SF(\xi)]) < 0$. The case where $Sup_R(\varphi')$ exists is similar. $\square$

This lemma will of course help to establish inductive proofs on top-down renamings, but we also need a more specific lemma concerning disjunctions.

LEMMA 3.2. *Let $\varphi \sqsubseteq \psi$ with $\varphi = \varphi_1 \vee \cdots \vee \varphi_n$ and $a_\varphi^\psi = 1$, R a renaming of φ top-down in ψ, and $S = \{i \in \{1..n\} / p(\varphi_i) > 1\}$, then $\exists k \in S, \varphi_k \notin R \wedge \forall i \in S - \{k\}, \varphi_i \in R$.*

PROOF. It is very easy to see that if there are at least two φ_i's with $p(\varphi_i) > 1$, their benefit in ψ is positive, and negative otherwise. We conclude that they all are in R (from the saturation condition) except one (from the condition that R is free). □

We can now show our theorem of invariance:

THEOREM 3.1. *$\forall \varphi \sqsubseteq \psi$, let R, R' be two renamings of φ top-down in ψ, then*

$$p(Rnm(R, \psi)) = p(Rnm(R', \psi))$$

PROOF. We show by induction on φ, that the equality holds for any ψ, R, R'.

If φ is a literal, we have $R = R' = \emptyset$, and the equality is trivial. Otherwise, we first suppose that $\varphi = \psi$ or $\varphi \sqsubset \psi \wedge B(\varphi, \psi) < 0$, hence that $\varphi \notin R$, $\varphi \notin R'$ (since R, R' are free in ψ), and consider three cases:

1 if φ is a quantified formula $Qx_1..x_n.\varphi'$, then $B(\varphi', \psi) = B(\varphi, \psi) < 0$, hence $\varphi' \notin R$, $\varphi' \notin R'$, and from lemma 3.1, R and R' are renamings of φ' top-down in ψ. The induction hypothesis for $\varphi' \sqsubset \varphi$ yields $p(Rnm(R, \psi)) = p(Rnm(R', \psi))$.

2 if φ is a conjunction $\varphi_1 \wedge \cdots \wedge \varphi_n$, let $R_k = R \cap SF^*(\varphi_k)$ and $R'_k = R' \cap SF^*(\varphi_k)$. Since $B(\varphi_k, \psi) = B(\varphi, \psi) < 0$, we infer from lemma 3.1 that R_k (resp. R'_k) is a renaming of φ_k top-down in $\psi[R - R_k]$ (resp. $\psi[R - R'_k]$). But the replacement of subformulas which are not in $SF^*(\varphi_k)$ is actually irrelevant to the benefits of subformulas of φ_k, because of the conjunction. Hence for any $R''_k \subset SF^*(\varphi)$, such that $R''_k \cap SF(\varphi_k) = \emptyset$, R_k and R'_k are actually top-down in $\psi[R''_k]$. This is true if we take $R''_k = \bigcup_{i=1}^{k-1} R'_i \cup \bigcup_{i=k+1}^{n} R_i$, with which we can apply the induction hypothesis for each φ_k, yielding $p(Rnm(R_k, \psi[R''_k])) = p(Rnm(R'_k, \psi[R''_k]))$, which can then be translated into

$$P_\varphi^\psi(\sum_{i=1}^{k-1} p(\varphi_i[R'_i]) + \sum_{i=k}^{n} p(\varphi_i[R_i])) + r_k = P_\varphi^\psi(\sum_{i=1}^{k} p(\varphi_i[R'_i]) + \sum_{i=k+1}^{n} p(\varphi_i[R_i])) + r'_k$$

with $r_k = \sum_{\varphi' \in R_k} p(Def_\psi(\varphi'[R_k]))$ and $r'_k = \sum_{\varphi' \in R'_k} p(Def_\psi(\varphi'[R'_k]))$. Using these n equalities we have

$$\begin{aligned}
& p(Rnm(R, \psi)) \\
&= P_\varphi^\psi(\sum_{i=1}^{n} p(\varphi_i[R_i])) + \sum_{i=1}^{n} r_i \\
&= P_\varphi^\psi(\sum_{i=1}^{k} p(\varphi_i[R'_i]) + \sum_{i=k+1}^{n} p(\varphi_i[R_i])) + \sum_{i=1}^{k} r'_i + \sum_{i=k+1}^{n} r_i \text{ (by induction on } k) \\
&= P_\varphi^\psi(\sum_{i=1}^{n} p(\varphi_i[R'_i])) + \sum_{i=1}^{n} r'_i \text{ (with } k = n) \\
&= p(Rnm(R', \psi))
\end{aligned}$$

3 if φ is a disjunction $\varphi_1 \vee \cdots \vee \varphi_n$, and $p(\varphi) > 1$ (the case $p(\varphi) = 1$ is trivial since then $R = R' = \emptyset$), then $a^{\psi}_{\varphi} = 1$ (because $B(\varphi, \psi) < 0$), and we are in the conditions of lemma 3.2. Let $S = \{i \in \{1..n\} \,/\, p(\varphi_i) > 1\}$, and $k, k' \in S$ such that $\varphi_k \notin R, \varphi_{k'} \notin R'$, and let $R_i = R \cap SF^*(\varphi_i), R'_i = R' \cap SF^*(\varphi_i)$. Splitting R in two, we have

$$p(Rnm(R,\psi)) = p(Rnm(R_k, \psi[R - R_k])) + \sum_{\xi \in R - R_k} p(Def_\psi(\xi[R]))$$

but $R - R_k$ is the disjoint union of the $R_i \cup \{\varphi_i\}$ for $i \in S - \{k\}$, and we clearly have

$$\sum_{\xi \in R_i \cup \{\varphi_i\}} p(Def_\psi(\xi[R])) = p(\varphi_i[R]) + \sum_{\xi \in R_i} p(Def_{\varphi_i}(\xi[R_i])) = p(Rnm(R_i, \varphi_i))$$

since ψ is in negation normal form. Doing the same work on R', we obtain

$$\begin{cases} p(Rnm(R,\psi)) = p(Rnm(R_k, \psi[R - R_k])) + \sum_{i \in S - \{k\}} p(Rnm(R_i, \varphi_i)) \\ p(Rnm(R',\psi)) = p(Rnm(R'_{k'}, \psi[R' - R'_{k'}])) + \sum_{i \in S - \{k'\}} p(Rnm(R'_i, \varphi_i)) \end{cases}$$

For $i \neq k, k'$, we have $\varphi_i \in R \cap R'$, hence R_i and R'_i are renamings of φ_i top-down in φ_i (lemma 3.1), hence $p(Rnm(R_i, \varphi_i)) = p(Rnm(R'_i, \varphi_i))$ (induction hypothesis). From this we obtain

$$\begin{aligned} p(Rnm(R,\psi)) - p(Rnm(R',\psi)) &= p(Rnm(R_k, \psi'[\varphi_{k'}])) - p(Rnm(R'_k, \varphi_k)) \\ &\quad - p(Rnm(R'_{k'}, \psi'[\varphi_k])) + p(Rnm(R_{k'}, \varphi_{k'})) \end{aligned}$$

where $\psi' = \psi[\varphi_i]_{i \in S - \{k,k'\}}$ (as $\varphi_k \notin R$, we have $\psi[R - R_k] = \psi[\varphi_i]_{i \in S - \{k\}}$). However, it is very much the same to rename subformulas of φ_k inside $\psi'[\varphi_{k'}]$ or inside φ_k, because the benefits are the same (φ_k is the only non atomic disjunct in $\psi'[\varphi_{k'}]$). We deduce that R_k (resp. R'_k) is top-down in φ_k (resp. $\varphi_{k'}$), and that

$$\begin{aligned} p(Rnm(R,\psi)) - p(Rnm(R',\psi)) &= p(Rnm(R_k, \varphi_k)) - p(Rnm(R'_k, \varphi_k)) \\ &\quad - p(Rnm(R'_{k'}, \varphi_{k'})) + p(Rnm(R_{k'}, \varphi_{k'})) \\ &= 0 \end{aligned}$$

from the induction hypothesis for φ_k and $\varphi_{k'}$.

We now suppose that $\varphi \sqsubset \psi$ and $B(\varphi, \psi) \geq 0$, hence from the saturation condition, that $\varphi \in R \cap R'$, and we therefore have $Sup_R(\varphi) = Sup_{R'}(\varphi) = \varphi$. Since $R, R' \subset SF(\varphi)$, we have $R \cap SF^*(\varphi) = R - \{\varphi\}$ and $R' \cap SF^*(\varphi) = R' - \{\varphi\}$, and from lemma 3.1 these are renamings of φ top-down in $Def_\psi(\varphi)$. As $B(\varphi, Def_\psi(\varphi)) < 0$, we proved in the previous case that $p(Rnm(R - \{\varphi\}, Def_\psi(\varphi))) = p(Rnm(R' - \{\varphi\}, Def_\psi(\varphi)))$. But $p(Rnm(R,\psi)) = p(\psi[\varphi]) + p(Rnm(R - \{\varphi\}, Def_\psi(\varphi)))$, and the same holds for R', hence we also have $p(Rnm(R,\psi)) = p(Rnm(R',\psi))$. □

3.3. OPTIMALITY

It is clear from this theorem of invariance that in order to show that a particular top-down renaming $R_{inf}(\psi)$ is optimal, it is sufficient to show that *there exists* an optimal top-down renaming. To prove this, we are going to use a well-founded preorder on renamings, decreasing from any optimal renaming to a top-down one. Let us call the ***depth*** of $\varphi \sqsubseteq \psi$

in ψ, and note $d_\psi(\varphi)$, the cardinal of $\{\varphi' \sqsubseteq \psi / \varphi \sqsubset \varphi'\}$. We extend this definition to renamings R by: $d_\psi(R) = \sum_{\varphi \in R} d_\psi(\varphi)$, which gives a preorder defined by $R \prec R'$ iff $d_\psi(R) \leq d_\psi(R')$. As $\leq$ on $\mathbb{N}$ is well-founded, there is no infinite sequence of renamings $(R_i)_{i \in \mathbb{N}}$ such that $\forall i \in \mathbb{N}, R_{i+1} \prec R_i \wedge R_i \not\prec R_{i+1}$.

We already know that any optimal renaming of ψ is free in ψ, hence an optimal renaming R which is not top-down in ψ does not satisfy the saturation condition, and this fact will enable us to construct $R' \prec R$ (such that $R \not\prec R'$). We will also have to restrict ourselves to complete renamings, which is a problem since an optimal renaming R is not necessarily complete. But it is easy to obtain a complete and optimal renaming form R, just by adding the subformulas of benefit 0. We can now prove the following

THEOREM 3.2. *Let R be an optimal, complete renaming of ψ non top-down in ψ, there exists an optimal and complete renaming R' of ψ such that $R' \prec R$ and $R \not\prec R'$.*

PROOF. The saturation condition does not hold, hence $\exists \varphi \sqsubset \psi, B(\varphi, \psi'[R - SF(\varphi)]) \geq 0$ with $\psi' = Def_\psi(Sup_R(\varphi))$ if defined, and $\psi' = \psi$ otherwise. Hence $\varphi \notin R$, otherwise $\varphi = Sup_R(\varphi)$, and this benefit would be negative. We also have $Inf_R(\varphi) \neq \emptyset$, otherwise we would have $R = R - SF(\varphi)$, and $B(\varphi, \psi'[R]) \geq 0$, in contradiction with R complete. Let $\varphi' \in Inf_R(\varphi)$, and $R' = R - \{\varphi'\} \cup \{\varphi\}$. Obviously, $\varphi' \sqsubset \varphi$, thus $R' \prec R, R \not\prec R'$.

From the definition of R' we can infer, first, that $\psi'[R' - SF(\varphi)] = \psi'[R - SF(\varphi)]$ (since $\varphi' \sqsubset \varphi$), hence $P_\varphi^{\psi'[R']} = P_\varphi^{\psi'[R]}$; next that $\varphi[R' - SF(\varphi')] = \varphi[R - SF(\varphi')]$, hence $P_{\varphi'}^{\varphi[R']} = P_{\varphi'}^{\varphi[R]}$; and last that $\varphi'[R'] = \varphi'[R]$. Therefore

$$\begin{aligned}
&p(Rnm(R,\psi)) - p(Rnm(R',\psi)) \\
&\quad = p(\psi'[R]) - p(\psi'[R']) + p(Def_\psi(\varphi'[R])) - p(Def_\psi(\varphi[R'])) \\
&\quad = P_\varphi^{\psi'[R]}(P_{\varphi'}^{\varphi[R]}(1)) - P_\varphi^{\psi'[R']}(1) + p(\varphi'[R]) - P_{\varphi'}^{\varphi[R']}(p(\varphi'[R'])) \\
&\quad = P_\varphi^{\psi'[R]}(P_{\varphi'}^{\varphi[R]}(1)) - P_\varphi^{\psi'[R]}(1) + p(\varphi'[R]) - P_{\varphi'}^{\varphi[R]}(p(\varphi'[R]))
\end{aligned}$$

But $B(\varphi, \psi'[R - SF(\varphi)]) \geq 0$, thus $a_\varphi^{\psi'[R]} > 1$, and $B(\varphi, \psi'[R]) < 0$, hence $p(\varphi[R]) = 1$, which shows that $P_{\varphi'}^{\varphi[R]} = Id$. We conclude that $p(Rnm(R,\psi)) = p(Rnm(R',\psi))$, and that R' is optimal.

There remains to establish that R' is complete, which is similar: $\forall \xi \sqsubset \psi$, let $e = B(\xi, Rnm(R',\psi))$

1 if $Sup_{R'}(\xi) \notin \{\varphi, \psi'\}$ (if $Sup_{R'}(\xi)$ does not exist, the condition is $\psi' \neq \psi$), the change between φ and φ' has no influence: $e = B(\xi, Rnm(R,\psi)) < 0$ since R is complete in ψ.

2 if $Sup_{R'}(\xi) = \psi'$ (or $\psi' = \psi$ if $Sup_{R'}(\xi)$ does not exist), then $e = B(\xi, \psi'[R']) = B(\xi, \psi'[R])$ since $p(\varphi[R']) = 1$. Hence $e = B(\xi, Rnm(R,\psi)) < 0$.

3 if $Sup_{R'}(\xi) = \varphi$, then either $\xi \sqsubseteq \varphi'$, and $e = B(\xi, Rnm(R,\psi))$ since $P_{\varphi'}^{\varphi[R']} = Id$; or $\varphi' \sqsubseteq \xi$, and $P_\xi^{\varphi[R]} = Id$, hence $e = B(\xi, \varphi[R]) < 0$; or else ξ and φ' are disjoint, and $\xi[R'] = \xi[R] \sqsubseteq \varphi[R]$, with $p(\varphi[R]) = 1$, thus $p(\xi[R']) = 1$ and $e < 0$.

□

From the existence of an optimal, hence of an optimal and complete renaming we then deduce the existence of an optimal and top-down renaming, hence that ***all top-down renamings are optimal.***

3.4. AN ALGORITHM FOR A TOP-DOWN SEARCH

We now have to show that with a particular top-down search, the procedure R_{inf} computes a top-down renaming. We first give an algorithm computing R_{inf} with the depth-first left-right top-down search. It works in two passes on the formula ψ considered as a labelled tree. The first pass consists in computing the values $p(\varphi)$ and $\overline{p}(\varphi)$ for each $\varphi \sqsubseteq \psi$, storing them as attributes; respectively $\varphi.p$ and $\varphi.\overline{p}$. This is easily programmed following table 1. The second pass is performed by calling $R\_rec(\psi, 1, 0, 1)$.

$R\_rec(\varphi, a, b, s) =$
  if $\langle \varphi.p, \varphi.\overline{p} \rangle \neq \langle 1, 1 \rangle$ **then**
    if $a * \varphi.p + b * \varphi.\overline{p} \geq a + b + if\_pos(s, \varphi.p) + if\_pos(-s, \varphi.\overline{p})$ **then**
      begin
        $R := R \cup \{\varphi\}$;
        $R\_rec(\varphi, if\_pos(s, 1), if\_pos(-s, 1), s)$;
        $\langle \varphi.p, \varphi.\overline{p} \rangle := \langle 1, 1 \rangle$
      end
    else let $\langle \varphi_1 .. \varphi_n \rangle = SFd(\varphi)$ **in**
      begin
        for $i := 1$ **to** n **do** $R\_rec(\varphi_i, a_{\varphi_i}^{Rnm(R,\psi)}, b_{\varphi_i}^{Rnm(R,\psi)}, s * pol(\varphi_i, \varphi))$;
        $\langle \varphi.p, \varphi.\overline{p} \rangle := nbcl(\varphi)$
      end

$if\_pos(x, y) =$ **if** $x \geq 0$ **then** y **else** 0

where $a_{\varphi_i}^{Rnm(R,\psi)}$ and $b_{\varphi_i}^{Rnm(R,\psi)}$ are computed from $a = a_{\varphi}^{Rnm(R,\psi)}$, $b = b_{\varphi}^{Rnm(R,\psi)}$, $\varphi_i.p$ and $\varphi_i.\overline{p}$ according to table 2. $nbcl(\varphi)$ computes $p(\varphi), \overline{p}(\varphi)$ from the attributes $\varphi'.p, \varphi'.\overline{p}$ of the direct subformulas φ' of φ, the list of these subformulas being computed by $SFd(\varphi)$. The worst case complexity of this algorithm is $O(|SF(\psi)|^2)$, one factor coming from the search, and the other from the arithmetic computations (bounded by $\log p(\psi) \leq |SF(\psi)|$).

It is obviously not difficult to compute $Rnm(R, \psi)$ once R is computed, but it is also possible to skip this step and compute directly the result of the renaming transformation. This requires a slight change in the search since we then need the free variables of the subformulas in order to build their definitions: if $\varphi.p = \varphi.\overline{p} = 1$, we have to search φ for its free variables. The complexity is then $O(|SF(\psi)||\psi|)$.

From a practical point of view, the program is based on arithmetic computations and pointer manipulations, and is very efficient. It is implemented in Common-Lisp on a SUN3 workstation, and is part of our interactive formula transformer, briefly presented in (Boy de la Tour *et al.*, 1988).

We call $R_{inf}^{\downarrow}$ the function computed by this algorithm. Its correction with respect to R_{inf} is rather obvious. We will also admit in the following that the subformulas are search according to the order $\lhd$, defined by $\varphi \lhd \varphi'$ iff either $\varphi' \sqsubseteq \varphi$ or φ and φ' are disjoint and, if $\langle \varphi_1 .. \varphi_n \rangle = SFd(\varphi \sqcup \varphi')$ with $\varphi \sqsubseteq \varphi_j$ and $\varphi' \sqsubseteq \varphi_{j'}$, then $j < j'$. There remains to show that $R_{inf}^{\downarrow}(\psi)$ is top-down in ψ.

The saturation condition is rather obvious, but we have to prove that $R^{\downarrow}_{inf}(\psi)$ is free in ψ. In other words, we have to show that whatever the order of renamings, the benefits are positive. We first prove a lemma of independence concerning disjoint subformulas (within top-down saturation).

Lemma 3.3. *$\forall \varphi, \varphi' \sqsubset \psi$, if φ and φ' are disjoint and such that $\forall \xi \sqsubset \psi, \varphi \sqsubset \xi \Rightarrow B(\xi, \psi) < 0$, then $B(\varphi, \psi) \geq 0 \wedge B(\varphi', \psi[\varphi]) \geq 0 \Rightarrow B(\varphi, \psi[\varphi']) \geq 0$.*

Proof. Let $\varphi \sqcup \varphi'$ be the smaller supformula of φ and φ' (lub of φ, φ' in the order $\sqsubseteq$). If $\varphi \sqcup \varphi'$ is a conjunction, we have $B(\varphi, \psi[\varphi']) = B(\varphi, \psi) \geq 0$, otherwise $\varphi \sqcup \varphi'$ is a disjunction $\varphi_1 \vee \cdots \vee \varphi_n$, with $\varphi \sqsubseteq \varphi_j, \varphi' \sqsubseteq \varphi_{j'}$ $(j \neq j')$. We have $B(\varphi \sqcup \varphi', \psi) < 0$ (since $\varphi \sqsubset \varphi \sqcup \varphi'$) and $p(\varphi \sqcup \varphi') > 1$, hence $a^{\psi}_{\varphi \sqcup \varphi'} = 1$.

We now suppose that $B(\varphi, \psi[\varphi']) < 0$. Let $b, b', c \in \mathbb{N}$ such that $P^{\varphi_j}_{\varphi}(x) = a^{\varphi_j}_{\varphi} x + b$, $P^{\varphi_{j'}}_{\varphi'}(x) = a^{\varphi_{j'}}_{\varphi'} x + b'$, $P^{\psi}_{\varphi \sqcup \varphi'}(x) = x + c$, and let $d = \prod_{i \neq j, j'} p(\varphi_i)$. We have $P^{\psi[\varphi']}_{\varphi}(x) = (a^{\varphi_j}_{\varphi} x + b)(a^{\varphi_{j'}}_{\varphi'} + b')d + c$, and since $p(\varphi) > 1$, $a^{\psi[\varphi']}_{\varphi} = (a^{\varphi_{j'}}_{\varphi'} + b') a^{\varphi_j}_{\varphi} d = 1$, hence $a^{\varphi_j}_{\varphi} = a^{\varphi_{j'}}_{\varphi'} = d = 1$ and $b' = 0$. Thus $P^{\psi[\varphi]}_{\varphi'}(x) = (1+b)x + c$, but $B(\varphi', \psi[\varphi]) \geq 0$, hence $b > 0$, and $\varphi \sqsubset \varphi_j$. We conclude that $B(\varphi_j, \psi) < 0$.

However, we have $p(\varphi_j) = p(\varphi) + b > 1$ and $a^{\psi}_{\varphi_j} = a^{\varphi \sqcup \varphi'}_{\varphi_j} = p(\varphi_{j'}) = p(\varphi') > 1$ since $B(\varphi', \psi[\varphi]) \geq 0$. Hence $B(\varphi_j, \psi) \geq 0$, and we have a contradiction; we conclude that $B(\varphi, \psi[\varphi']) \geq 0$. □

The next lemma establishes an analogous result of independence for the complementary case, that is for non disjoint subformulas.

Lemma 3.4. *$\forall \varphi' \sqsubseteq \varphi \sqsubset \psi, B(\varphi, \psi) \geq 0 \wedge B(\varphi', Def_{\psi}(\varphi)) \geq 0 \Rightarrow B(\varphi, \psi[\varphi']) \geq 0$*

Proof. We have $a^{\psi}_{\varphi} > 1$ and $a^{Def_{\psi}(\varphi)}_{\varphi'} = a^{\varphi}_{\varphi'} > 1$, hence $p(\varphi[\varphi']) = P^{\varphi}_{\varphi'}(1) \geq a^{\varphi}_{\varphi'} > 1$. But $a^{\psi[\varphi']}_{\varphi} = a^{\psi}_{\varphi} > 1$, hence $B(\varphi, \psi[\varphi']) \geq 0$. □

From these two lemmas we can prove that

Theorem 3.3. *$\forall \psi, R^{\downarrow}_{inf}(\psi)$ is top-down in ψ.*

Proof. Let $R = R^{\downarrow}_{inf}(\psi)$ and $\{\varphi_1..\varphi_n\} = R$ such that $\varphi_1 \lhd \cdots \lhd \varphi_n$. It is easy to show that the saturation condition holds: $\forall \varphi \sqsubseteq \psi$, let i such that $\varphi_i \lhd \varphi$ and $\varphi_{i+1} \not\lhd \varphi$ (if $\varphi_1 \not\lhd \varphi$ let $i = 0$), we have $B(\varphi, \psi'[\varphi_1..\varphi_i]) < 0$, with $\psi' = Def_{\psi}(Sup_R(\varphi))$ if defined, and $\psi' = \psi$ otherwise. But $SF(\varphi) \cap \{\varphi_1..\varphi_i\} = \emptyset$, hence $B(\varphi, \psi'[R - SF(\varphi)]) < 0$ from *ftm*.

We now have to establish that R is free in ψ. Let $i \in \{1..n\}$, and if $\exists \varphi_m \in R$ such that $\varphi_i \in Inf_R(\varphi_m)$, let $\psi_i = Def_{\psi}(\varphi_m[\varphi_1..\varphi_{i-1}])$, otherwise let $\psi_i = \psi[\varphi_1..\varphi_{i-1}]$. We clearly have $B(\varphi_i, Rnm(R - \{\varphi_i\}, \psi)) = B(\varphi_i, \psi_i[\varphi_{i+1}..\varphi_n])$, which we have to prove positive. Up to now, we only have, by definition of R_{inf}:

$$\forall j \in \{1..n\}, B(\varphi_j, Rnm(\{\varphi_1..\varphi_{j-1}\}, \psi)) \geq 0 \tag{3.1}$$

Let p such that $\varphi_{i+1}..\varphi_p \sqsubset \varphi_i$ and $\varphi_{p+1}..\varphi_n \not\sqsubseteq \varphi_i$. For $j = i..p$, we consider the hypothesis $B(\varphi_i, \psi_i[\varphi_{i+1}..\varphi_j]) \geq 0$, which is true for $j = i$, according to (3.1). If $\varphi_{j+1} \in Inf_R(\varphi_i)$, we have from (3.1) $B(\varphi_{j+1}, Def_\psi(\varphi_i[\varphi_{i+1}..\varphi_j])) \geq 0$, hence from lemma 3.4, together with our hypothesis, we obtain $B(\varphi_i, \psi_i[\varphi_{i+1}..\varphi_{j+1}]) \geq 0$. Otherwise, we have $\exists \varphi_k \in R$ such that $\varphi_{j+1} \in Inf_R(\varphi_k)$ and $\varphi_k \sqsubset \varphi_i$, hence $\varphi_{i+1} \lhd \varphi_k \lhd \varphi_j$, thus $k \in \{i+1..j\}$, and $\psi_i[\varphi_{i+1}..\varphi_{j+1}] = \psi_i[\varphi_{i+1}..\varphi_j]$. By induction on j, we conclude that the hypothesis is true; in particular for $j = p$ we have $B(\varphi_i, \psi_i') \geq 0$ with $\psi_i' = \psi_i[\varphi_{i+1}..\varphi_p]$.

For $j = p..n$, we now consider the hypothesis $B(\varphi_i, \psi_i'[\varphi_{p+1}..\varphi_j]) \geq 0$, which is true for $j = p$, as shown above. If $\varphi_{j+1} \in Inf_R(\varphi_m)$ (if φ_m is undefined, the condition is $\varphi_{j+1} \in Inf_R(\psi)$), we have from (3.1) $B(\varphi_{j+1}, \psi_i'[\varphi_i\varphi_{p+1}..\varphi_j]) \geq 0$, hence from lemma 3.3, together with the hypothesis, we obtain $B(\varphi_i, \psi_i'[\varphi_{p+1}..\varphi_{j+1}]) \geq 0$. Otherwise, we have as above $\psi_i'[\varphi_{p+1}..\varphi_{j+1}] = \psi_i'[\varphi_{p+1}..\varphi_j]$. By induction on j, the hypothesis is true, and we conclude with $j = n$ that $B(\varphi_i, \psi_i[\varphi_{i+1}..\varphi_n]) \geq 0$. □

It should be clear that this theorem actually holds for $R_{inf}(\psi)$ with any top-down search (the proof is simpler with $R_{inf}^{\downarrow}(\psi)$), hence the notion of top-down renaming corresponds exactly to the renamings obtained from the procedure R_{inf} with a top-down search. The correspondence only holds for linear formulas; example 3.2 exhibits a non free (hence non top-down) renaming computed by means of a top-down search. For a renaming, being free seems to be the key property to optimality — it is an open problem whether there exists a non optimal renaming of ψ free in ψ.

3.5. AN OPTIMIZATION

One problem with renamings is that they tend to prevent simplifications:

EXAMPLE 3.3. Let $\psi = \varphi \vee (\neg A \wedge \neg B)$ with $\varphi = A \wedge B$, the conjunctive normal form of ψ contains the two tautologies $A \vee \neg A$ and $B \vee \neg B$, hence simplifies to $(A \vee B) \wedge (\neg A \vee \neg B)$. However, $R_{inf}^{\downarrow}(\psi) = \{\varphi\}$, and the corresponding clause form is $(SkP_\varphi^\psi \vee \neg A) \wedge (SkP_\varphi^\psi \vee \neg B) \wedge (\neg SkP_\varphi^\psi \vee A) \wedge (\neg SkP_\varphi^\psi \vee B)$, which cannot be simplified, and contains four clauses instead of two.

Hence it is clear that we have to avoid the renaming of subformulas which are not strictly necessary. According to our criterion of renaming, it does not seem to be indispensable to include subformulas of benefit zero. But our proofs strongly rely on their inclusion. It is however easy to transpose the results of section 2 to the *strictly positive* renamings, that is renamings $\{\varphi_1..\varphi_n\}$ corresponding to sequences $\varphi_1..\varphi_n$ such that $\forall i \in \{1..n\}, B(\varphi_i, Rnm(\{\varphi_1..\varphi_{i-1}\}, \psi)) > 0$, and *almost complete*, that is verifying $\forall \varphi \sqsubset \psi, B(\varphi, Rnm(R, \psi)) \leq 0$. The main point is that from any strictly positive and almost complete renaming R it is possible to build a positive and complete renaming $R' \supset R$ ($R' - R$ contains subformulas of benefit zero) such that $p(Rnm(R, \psi)) = p(Rnm(R', \psi))$, hence theorem 2.2 still holds, and the results of complexity follow.

However, this operation, from R to R', does not preserve the search, and cannot be applied to the results of the present section. It may seem evident that replacing the test $B(\varphi, \psi) \geq 0$ by $B(\varphi, \psi) > 0$ brings fewer clauses, but this is not the case. It first tends to prevent renamings, but this may increase the subsequent benefits, according to *ftm*, hence add other subformulas. At the end, we may obtain a very different renaming, which seems to be the case on non linear formulas, and the number of clauses is then very difficult to compare. However, restricting ourselves once again to linear formulas, and to the search adopted in $R^{\downarrow}_{inf}$, we can prove that the renaming obtained from ψ by replacing $\geq$ by $>$ in our algorithm, which we call $R_{opt}(\psi)$, is included in $R^{\downarrow}_{inf}(\psi)$.

Theorem 3.4. $\forall\psi, R_{opt}(\psi) \subset R^{\downarrow}_{inf}(\psi)$

Proof. $\forall\xi \sqsubset \psi$, let $R_\varphi = \{\xi \in R_{opt}(\psi) / \xi \lhd \varphi\}$ and $R'_\varphi = \{\xi \in R^{\downarrow}_{inf}(\psi) / \xi \lhd \varphi\}$, we show by induction on φ in the order $\lhd$ that $R_\varphi \subset R'_\varphi$, our induction hypothesis being $\forall\xi \sqsubset \psi, \xi \lhd \varphi \wedge \xi \neq \varphi \Rightarrow R_\xi \subset R'_\xi$. Since $R_\xi - \{\xi\}$ corresponds to $R_{\xi'}$ where ξ' is the $\lhd$-predecessor of ξ, we then have $\forall\xi \lhd \varphi, R_\xi - \{\xi\} \subset R'_\xi - \{\xi\}$.

We first show, as a lemma, that:

$$\forall\xi \sqsubset \psi, \xi \in R'_\varphi - R_\varphi \Rightarrow a_\xi^{Rnm(R_\xi,\psi)} = a_\xi^{Rnm(R'_\xi-\{\xi\},\psi)} = p(\xi) = 2 \tag{3.2}$$

If $\xi \in R'_\varphi - R_\varphi$, then from the definition of $R^{\downarrow}_{inf}$ (resp. R_{opt}) $B(\xi, Rnm(R'_\xi - \{\xi\}, \psi)) \geq 0$ (resp. $B(\xi, Rnm(R_\xi, \psi)) \leq 0$), hence $a_\xi^{Rnm(R'_\xi-\{\xi\},\psi)} > 1$ and $p(\xi[R'_\xi]) > 1$. By definition of $\lhd$ we have $R'_\xi \cap SF^*(\xi) = \emptyset$, hence $p(\xi) > 1$. We also have $(a_\xi^{Rnm(R_\xi,\psi)} - 1)(p(\xi) - 1) \leq 1$ since $R_\xi \cap SF^*(\xi) = \emptyset$. But $R_\xi - \{\xi\} = R_\xi \subset R'_\xi - \{\xi\}$ from the induction hypothesis ($\xi \lhd \varphi$ since $\xi \in R'_\xi$), hence from *ftm* $a_\xi^{Rnm(R_\xi,\psi)} \geq a_\xi^{Rnm(R'_\xi-\{\xi\},\psi)} > 1$, and the only possibility is then $a_\xi^{Rnm(R_\xi,\psi)} = p(\xi) = 2$, hence $a_\xi^{Rnm(R'_\xi-\{\xi\},\psi)} = 2$, and formula (3.2) is proved.

We suppose that $R_\varphi \not\subset R'_\varphi$, and try to find a contradiction. Since $R_\varphi - \{\varphi\} \subset R'_\varphi - \{\varphi\}$, then $\varphi \in R_\varphi - R'_\varphi$, hence $B(\varphi, Rnm(R_\varphi - \{\varphi\}, \psi)) > 0$ and $B(\varphi, Rnm(R'_\varphi, \psi)) < 0$. Let $\psi' = Sup_{R_\varphi - \{\varphi\}}(\varphi)$ if defined, and $\psi' = \psi$ otherwise, we have as above (with $R'_\varphi \cap SF(\varphi) = \emptyset$) $a_\varphi^{\psi'[R_\varphi]} > 1$, $p(\varphi) > 1$ and $a_\varphi^{Rnm(R'_\varphi,\psi)} = 1$.

But we can prove that $Sup_{R'_\varphi}(\varphi) = Sup_{R_\varphi - \{\varphi\}}(\varphi)$. If it is not true, let $\xi = Sup_{R'_\varphi}(\varphi)$, which is defined since $R_\varphi - \{\varphi\} \subset R'_\varphi$, we have $\xi \in R'_\varphi - R_\varphi$ and $\xi \lhd \varphi$, hence $a_\xi^{\psi'[R_\xi]} = p(\xi) = 2$ according to (3.2). We also have $a_\varphi^{Def_\psi(\xi[R'_\varphi])} = a_\varphi^{Rnm(R'_\varphi,\psi)} = 1$, hence $p(\xi[R'_\varphi]) = a_\varphi^{\xi[R'_\varphi]} p(\varphi) = p(\varphi)$, but $p(\xi[R'_\varphi]) \leq p(\xi) = 2$, hence $p(\varphi) = 2$. From $B(\varphi, Rnm(R_\varphi - \{\varphi\}, \psi)) > 0$, we have $a_\varphi^{\psi'[R_\varphi]} > 2$. But $a_\varphi^{\psi'[R_\varphi]} = a_\xi^{\psi'[R_\varphi]} a_\varphi^{\xi[R_\varphi]}$ with $a_\xi^{\psi'[R_\varphi]} = a_\xi^{\psi'[R_\xi]} = 2$ since $R_\varphi - R_\xi$ only contains subformulas of ξ (because $\varphi \sqsubset \xi$), and $a_\varphi^{\xi[R_\varphi]} \leq a_\varphi^\xi = 1$ since $p(\xi) = p(\varphi)$. Hence $a_\varphi^{\psi'[R_\varphi]} = 2$, and we have a contradiction, which proves that $Sup_{R'_\varphi}(\varphi) = Sup_{R_\varphi - \{\varphi\}}(\varphi)$.

We then have $a_\varphi^{\psi'[R'_\varphi]} = a_\varphi^{Rnm(R'_\varphi,\psi)} = 1$, thus $a_\varphi^{\psi'[R'_\varphi]} \neq a_\varphi^{\psi'[R_\varphi]}$, and therefore $Inf_{R_\varphi - \{\varphi\}}(\psi') \neq Inf_{R'_\varphi}(\psi')$. If $Inf_{R_\varphi - \{\varphi\}}(\psi') - Inf_{R'_\varphi}(\psi') \neq \emptyset$, let ξ be one of its element, we then have $\xi \in R_\varphi - \{\varphi\} \subset R'_\varphi$, i.e. $\xi \in R'_\varphi$ and $\xi \notin Inf_{R'_\varphi}(\psi')$, hence $\exists\xi' \in Inf_{R'_\varphi}(\psi')$ such that $\xi \sqsubset \xi'$, hence we also have $\xi' \notin R_\varphi$. Otherwise we have $Inf_{R'_\varphi}(\psi') - Inf_{R_\varphi - \{\varphi\}}(\psi') \neq \emptyset$, and let ξ be one of its element, if $\xi \in R_\varphi$ then $\exists\xi \in Inf_{R_\varphi - \{\varphi\}}(\psi')$ such that $\xi \sqsubset \xi'$, but then $\xi' \in R'_\varphi$, which is impossible since $\xi \in Inf_{R'_\varphi}(\psi')$; hence $\xi \notin R_\varphi$. In both cases we were able to find an element of $Inf_{R'_\varphi}(\psi') - R_\varphi$, which is therefore not empty.

In the hypothesis that $\forall\xi \in Inf_{R'_\varphi}(\psi') - R_\varphi, \xi \sqcup \varphi$ is a conjunction, the differences between $Inf_{R'_\varphi}(\psi')$ and $Inf_{R_\varphi - \{\varphi\}}(\psi')$ do not influence the benefit of φ: $B(\varphi, \psi'[R'_\varphi]) = B(\varphi, \psi'[R_\varphi - \{\varphi\}])$, which is impossible since the first is strictly negative and the second positive. Let $\varphi' \in Inf_{R'_\varphi}(\psi') - R_\varphi$ such that $\varphi' \sqcup \varphi$ is a disjunction $\varphi_1 \vee \cdots \vee \varphi_n$, and let i, j such that $\varphi' \sqsubseteq \varphi_i$ and $\varphi \sqsubseteq \varphi_j$ (we have $i < j$).

We also have $\varphi' \in R'_\varphi - R_\varphi$, hence $a_{\varphi'}^{\psi'[R_{\varphi'}]} = p(\varphi') = 2$ according to (3.2). But:

$$a_{\varphi'}^{\psi'[R_{\varphi'}]} = a_{\varphi' \sqcup \varphi}^{\psi'[R_{\varphi'}]} \prod_{k=1}^{i-1} p(\varphi_k[R_{\varphi'}]) a_{\varphi'}^{\varphi_i[R_{\varphi'}]} \prod_{k=i+1}^{j-1} p(\varphi_k) a_{\varphi}^{\varphi_j} p(\varphi) \prod_{k=j+1}^{n} p(\varphi_k)$$

and $p(\varphi) > 1$, hence $p(\varphi) = 2$ and the other factors are all equal to 1. In the same way:

$$a_{\varphi}^{\psi'[R_\varphi]} = a_{\varphi' \sqcup \varphi}^{\psi'[R_\varphi]} \prod_{k=1}^{i-1} p(\varphi_k[R_\varphi]) a_{\varphi'}^{\varphi_i[R_\varphi]} p(\varphi'[R_\varphi]) \prod_{k=i+1}^{j-1} p(\varphi_k[R_\varphi]) a_{\varphi}^{\varphi_j[R_\varphi]} \prod_{k=j+1}^{n} p(\varphi_k)$$

with $p(\varphi'[R_\varphi]) \le p(\varphi') = 2$, and as $R_{\varphi'} \subset R_\varphi$, all the other factors are less than in the previous equation, and equal 1. We then have $a_\varphi^{\psi'[R_\varphi]} \le 2$, thus $B(\varphi, \psi'[R_\varphi - \{\varphi\}]) \le 0$, which finally contradicts our hypothesis. We conclude that the induction is complete, hence that $R_{opt}(\psi) = R_\varphi \subset R'_\varphi = R^{\downarrow}_{inf}(\psi)$ where φ is the maximal subformula of ψ in the order $\lhd$. □

COROLLARY 3.1. *If ψ is linear, $R_{opt}(\psi)$ is optimal.*

PROOF. Since $R_{opt}(\psi)$ is almost complete, it is clear from theorems 2.1 and 3.4 that $p(Rnm(R_{opt}(\psi), \psi)) \le p(Rnm(R^{\downarrow}_{inf}(\psi), \psi))$, but $R^{\downarrow}_{inf}(\psi)$ is optimal according to theorems 3.1, 3.2 and 3.3, hence $R_{opt}(\psi)$ is optimal. □

4. Some Experimental Results

Of course, the fact that renamings tend to prevent simplifications does not make useless the remaining possible simplifications. But it is then all the more interesting to perform simplifications that are not affected by renamings, since they can be applied before renaming: these are *non-clausal* simplifications, coming from (Van Gelder, 1984) and extended to first order logic in a straightforward way. These simplifications are not very easy to implement, but they appear to be very efficient and useful — maybe more than clausal simplifications, although the comparison is difficult since the two techniques have different results; hence both should be used. For example, the theorems experimented in (Plaisted and Greenbaum, 1986) are all simplified to very simple formulas (most of them to the empty clause) which do not require particular renamings to be refuted (R_{opt} returns $\emptyset$ on these formulas). Significant comparisons between renamings can only be performed on more difficult examples.

In (Pelletier, 1986) can be found a problem that is designed to test clause form translators (problem 53), the expected number of clauses being 146. However, the polarity-dependent linearization (see section 2.1) alone, without any simplification or renaming, yields 34 clauses. Together with *struct_pres*, we obtain 41 clauses, but a much shorter length (466 instead of 1105). R_{opt} is very restricted: it contains only one subformula, and the resulting clause form has only 18 clauses, and also the shortest length (327 symbols).

A much more interesting and difficult theorem is Andrews's challenge problem:

$$\begin{aligned}\psi \quad = \quad & \neg\,[\exists x \forall y (P(x) \Leftrightarrow P(y)) \Leftrightarrow (\exists x Q(x) \Leftrightarrow \forall y P(y))] \\ & \Leftrightarrow [\exists x \forall y (Q(x) \Leftrightarrow Q(y)) \Leftrightarrow (\exists x P(x) \Leftrightarrow \forall y Q(y))]\end{aligned}$$

which yields thousands of clauses with the standard linearization (2704 clauses, length 79045). The polarity-dependent linearization also results in a quite big clause form: 128 clauses (length 3716). As far as we know, no resolution theorem prover is able to refute it; Andrews's problem has been solved with the resolution method only when ingeniously translated to clause form: see (Henschen *et al.*, 1980), (Guha and Zhang, 1989) and (Quaife, 1990). The structure preserving translation reduces the number of clauses to 43 (length 337); this was approximately the transformation in (Henschen *et al.*, 1980), together with simplifications performed by the TAMPR system, but not described in the paper, which reduced the number of clauses to 26 (length not specified). Using our resolution theorem prover on a SUN3-60 8MB, also described in (Boy de la Tour *et al.*, 1988), we found a refutation in 587 seconds. Following (Plaisted and Greenbaum, 1986), we adopted a predicate ordering strategy: skolem literals are resolved away last, and the bigger the renamed subformula, the later the corresponding skolem literal is resolved upon.

$R_{opt}(\psi)$ contains only three subformulas: the subformulas $\exists x \forall y (P(x) \Leftrightarrow P(y))$ and $\exists x \forall y (Q(x) \Leftrightarrow Q(y))$, and the first operand of the top-most $\Leftrightarrow$ sign. The resulting number of clauses is 24, with length 274 (obtained without simplifications, hence very efficiently). The clause form was refuted in 28.5 seconds. No shorter clause form is known presently; only in (Quaife, 1990) can be found another clause form with 24 clauses, also obtained with the renaming of 3 subformulas but with a bottom-up search, hence these subformulas are smaller and have more free variables. Hence the corresponding skolem literals also have more free variables, which increases the number of possible resolvents from these.

More intuitively, another drawback of renaming small subformulas instead of big ones is that it then seems more likely that some clausal simplifications are prevented. This is the case on a simpler version of Andrews's problem which is also presented in (Quaife, 1990): after Quaife's translation, there is only one possible simplification, which yields 20 clauses. Using R_{opt}, the resulting clause form, having the same number of unsimplified clauses, can be simplified to 16 clauses. This kind of behaviour is however likely to be highly problem dependent.

5. Conclusion

The choice we have made to focus on a tractable syntactic criterion, the number of clauses, has resulted in a very efficient translation into a concise clause form, with strong properties. Our more interesting results are certainly those concerning optimality: the restricted optimality, relative to the standard and the structure preserving translations, and the optimality of top-down renamings on linear formulas. These two positive results come with two negative ones: the non-optimality of top-down renamings in the general case, and a worst case complexity which is the complexity of the structure preserving translation multiplied by a factor n, that is the number of subformulas of the conjecture. However, experiments with difficult examples show that these two negative facts have little importance in comparison with the positive ones.

There are obviously other interesting criteria that can be used in order to rename a formula. One possibility considered in (Boy de la Tour, 1991) is to minimize the number of (occurrences of) literals; this is more complicated than just minimizing the number of clauses, but also very efficient. Although we can prove a fundamental theorem of monotony in this context, optimality of the number of literals does not hold uniformly on top-down strategies. The worst case complexity is better than with R_{opt}, but not equal to the one of *struct_pres* in first order logic. However, the main problem is that it seems to increase the number of renamings compared to R_{opt}.

There are also other ways to introduce definitions than what we have called renaming. It is actually possible to introduce a single definition for several occurrences of a single subformula, as suggested in (Plaisted and Greenbaum, 1986). This technique is obviously more difficult to automate than renaming, and it is difficult to know, for instance, whether it is interesting or not to introduce a definition for a subformula occurring only twice. This technique is used in (Bruschi, 1991), by hand, in order to decrease the number of clauses of a formulation of the halting problem from 86 to 43. Using R_{opt}, we obtain 33 clauses, showing that it may be difficult to compute an adequate clause form using this last technique for introducing definitions.

Let us finally emphasize on the fact that R_{opt} is very easy to implement, is compatible with any resolution theorem prover (at least if a literal ordering strategy is available, in order to delay the use of the new literals), has practically no computation cost, and seems worth it on many conjectures, if not indispensable when otherwise huge clause forms are considered; this is often the case when conjectures are produced automatically, for instance in program verification, or when translated from other logics.

Acknowledgements I thank Ricardo Caferra, Gilles Chaminade, Nicolas Zabel, and the anonymous referees for their helpful comments and careful reading of this paper.

References

Boy de la Tour, T. (1990). Minimizing the number of clauses by renaming. In *Proceedings of the 10th International Conference on Automated Deduction* (Stickel, M. E., ed.), pp. 558–572. Springer Lecture Notes in Artificial Intelligence 449.

Boy de la Tour, T. (1991). *Optimisations par renommage dans la méthode de résolution*. PhD thesis, Institut National Polytechnique de Grenoble.

Boy de la Tour, T., Caferra, R., and Chaminade, G. (1988). Some tools for an inference laboratory (atinf). In *Proccedings of the 9th International Conference on Automated Deduction* (Lusk, E. and Overbeek, R., eds), pp. 744–745. Springer Lecture Notes in Computer Science 310.

Boy de la Tour, T. and Chaminade, G. (1990). The use of renaming to improve the efficiency of clausal theorem proving. In *AIMSA'90, Artificial Intelligence—Methodology Systems Application* (Jorrand, P. and Sgurev, V., eds). North-Holland.

Bruschi, M. (1991). The halting problem. *AAR Newsletter*, **16**, 7–12.

Guha, A. and Zhang, H. (1989). Andrew's challenge problem: Clause conversion and solutions. *AAR Newsletter*, **14**, 5–8.

Henschen, L., Lusk, E., Overbeek, R., Smith, B., Veroff, R., Winker, S., and Wos, L. (1980). challenge problem 1. *SIGART newsletter*, pp. 30–31.

Pelletier, J. (1986). Seventy-five problems for testing automatic theorem provers. *Journal of Automated Reasoning*, **2**, 191–216.

Plaisted, D. A. and Greenbaum, S. (1986). A structure-preserving clause form translation. *Journal of Symbolic Computation*, **2**, 293–304.

Quaife, A. (1990). Andrew's challenge problem revisited. *AAR Newsletter*, **15**, 3–7.

Tseitin, G. S. (1968). On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic, Part II* (Slisenko, A. O., ed.).

Van Gelder, A. (1984). A satisfiability tester for non-clausal propositional calculus. In *Proceedings of the 7th International Conference on Automated Deduction* (Shostak, R., ed.), pp. 101–112. Springer Lecture Notes in Computer Science 170.